

EVALUACIÓN DE DIFERENTES PLATAFORMAS JAVA USANDO OPENCL

ALUMNO: RAQUEL MEDINA DOMÍNGUEZ

UNIVERSIDAD: REALIZADO EN UNIVERSITY OF SEOUL EN CONVENIO BILATERAL NO EUROPEO CON LA UNIVERSIDAD CARLOS III DE MADRID EN 1ER CUATRIMESTRE DE 2013

TUTORES

UC3M : JOSE MARIA SIERRA CAMARA sierra@inf.uc3m.es

UNIVERSITY OF SEOUL: YOUNGMIN YI ymyi@uos.ac.kr

INTRODUCCIÓN

Por una lado el CPU tradicional es capaz de ejecutar solo varios threads al mismo tiempo.

Por otro lado, la tecnologia GPU permite ejecutar cientos o miles de simples threads. La tecnologia GPU fue originalmente disenada para graficos pero en estos dias esta siendo usada para calculos usando GPGPU tecnologia. OpenCL, una de las GPGPU tecnologias, es introducida en este proyecto final.

- Metodos de cooperacion entre Java y el lenguaje presentado.

- Varias librerias son presentadas en este proyecto como Jogamp, JOCL y JavaCL.

- Mi propia clase multiplicacion de vectores ha sido usada como base de nuestra aplicacion para evaluar las diferentes librerias presentadas aqui.

Diferentes medidas han sido usadas para evaluar estas diferentes plataformas como son velocidad, tiempo y memoria usada. Acorde a estas medidas somos capaces de definir que tipo de libreria es mas adecuada para los diferentes proyectos que se deseen elaborar.

CONCLUSIONES (VER GRÁFICOS EN [finalversion.pdf](#))

Tras realizar diferentes pruebas sobre las librerías y clases implementadas que se pueden ver en el proyecto escrito en inglés llegamos a varias conclusiones.

Primero, diferenciamos entre las características de cada librería.

JoCl y JOGAMP son las librerías que muestran mejores resultados sobre memoria, aunque debemos admitir que son las más difíciles para trabajar con ellas.

JavaCL por otro lado es la librería más fácil para trabajar ya que como hemos visto podemos realizar el mismo trabajo con tres veces menos código, la documentación es más fácil de entender, y ha sido la única librería capaz de instalarse a través del terminal.

Segundo, los test sobre el tiempo.

La librería JOGAMP mostró ser la más efectiva. Fue la más rápida en orden de ejecución. JoCl y JavaCL obtuvieron resultados similares aunque JoCl fue un poco más efectiva pero siendo dos veces más lenta que JOGAMP.

Finalmente, los test de memoria.

JoCl fue la librería más efectiva, fue la librería que utilizó menos memoria utilizando los mismos datos que las otras dos librerías.

Además, JOGAMP and JavaCL obtuvieron el mismo resultado, usando tres veces más memoria que JoCl.

Como resultado final se observa que trabajando con Java y OpenCL, el programador que se adentre en un nuevo proyecto debe tener en cuenta estas características dependiendo de el resultado que quiera obtener, es decir buscando rapidez o eficiencia en memoria.

Siendo para todo ello la librería JOGAMP la más eficiente en rapidez, JoCl la más eficiente en memoria y siendo JavaCL la librería más simple de usar.